STARTING APP

# ---------------------------------------------------------------------------- #

# ACTIVITY 1.4B BARPLOT\_1 SERVER

# ---------------------------------------------------------------------------- #

# Load in the libraries

library(shiny)

library(dplyr)

library(ggplot2)

library(RColorBrewer)

# Load in the data

raw\_data <- read.csv("data/raw\_data.csv", stringsAsFactors=FALSE)

# Create a colour palette

col\_palette <- brewer.pal(name="Dark2", n=8)

#------------------------------------------------------------------------------#

# Begin server section

shinyServer(function(input, output) {

# Produce plot

output$barPlot <- renderPlot({

ggplot() +

geom\_bar(data = raw\_data,

aes\_string(x = input$xaxis),

fill = col\_palette[1]) +

labs(x = input$xaxis, y="Number of records") +

theme\_classic() + # use a simple theme (no gridlines)

theme(axis.text = element\_text(size=14),

axis.title = element\_text(size=18),

plot.title = element\_text(size=20),

legend.title = element\_text(size=18),

legend.text = element\_text(size=14))

}) # Close the renderPlot

}) # Close the shinyServer

STARTING APP

# ---------------------------------------------------------------------------- #

# ACTIVITY 1.4B BARPLOT\_1 UI

# ---------------------------------------------------------------------------- #

# Load in libraries

library(shiny)

#------------------------------------------------------------------------------#

# Begin ui section

shinyUI(fluidPage(

# Application title

titlePanel("Day 1 - Barplot\_1"),

# Add a line break

br(),

# Add text section

h4("This app our first introduction to rShiny!"),

h4("There is only 1 widget - a dropdown menu. This is set up to change the variable that is plotted on the x-axis."),

# Add a line break

br(),

# Create a sidebar layout

sidebarLayout(

# Add content to the Sidebar

sidebarPanel(

# Add a dropdown menu

selectInput(inputId="xaxis",

label = h3("Select the x-axis variable:"),

choices = list("Sex" = "sex", "Species" = "species"),

selected = 1)

), # Close the sidebarPanel

# Add content to the main panel

mainPanel(

# Show plot

plotOutput("barPlot",

height=700)

) # Close the mainPanel

) # Close the sidebarLayout

)) # Close the fluidPage and the ShinyUI

COMPLETED APP

# ---------------------------------------------------------------------------- #

# ACTIVITY 1.4B BARPLOT\_1 SERVER

# ---------------------------------------------------------------------------- #

library(shiny)

library(ggplot2)

raw\_data <- read.csv("data/raw\_data.csv", stringsAsFactors=FALSE)

# Create a colour palette

col\_palette <- brewer.pal(name="Dark2", n=8)

#------------------------------------------------------------------------------#

# Begin server section

shinyServer(function(input, output) {

# Produce plot

output$barPlot <- renderPlot({

ggplot() +

geom\_bar(data=raw\_data, aes(x=raw\_data[[input$xaxis]]), fill=col\_palette[1]) +

labs(x=paste0("\n", input$xaxis), y="Number of records\n") +

# Extra plotting code to control appearence

theme\_classic() +

theme(axis.text = element\_text(size=14),

axis.title = element\_text(size=18),

plot.title = element\_text(size=20),

legend.title = element\_text(size=18),

legend.text = element\_text(size=14))

})

})

COMPLETED APP

# ---------------------------------------------------------------------------- #

# ACTIVITY 1.4B BARPLOT\_1 UI

# ---------------------------------------------------------------------------- #

library(shiny)

#------------------------------------------------------------------------------#

# Begin ui section

shinyUI(fluidPage(

# Application title

titlePanel("Exploratory plots: Barplot\_1 (Master)"),

# Add a line break

br(),

# Add text section

h4("This app our first introduction to rShiny!"),

h4("There is only 1 widget: selectInput - a dropdown menu. We have set this up to change the variable that is plotted on the x-axis."),

# Add a line break

br(),

# Sidebar with a slider input for number of bins

sidebarLayout(

sidebarPanel(

# Add a dropdown menu

selectInput("xaxis", label = h3("Select the x-axis variable:"),

choices = list("Sex" = "sex", "Species" = "species", "Age" = "age"),

selected = 1)

),

# Show a plot of the generated distribution

mainPanel(

plotOutput("barPlot", height=700)

)

)

))